Автор: Лукащук Богдан

Веб-скрапінг. Частина 1

У даній частині буде розглянуто базову інформацію про веб-скрапінг, а також наведені приклади скрапінгу сайтів за допомогою бібліотек **Newspaper3k, Requests, BeautifulSoup.** Приклади коду з важливими коментарями наведені нижче за посиланням (та QR-кодом) на Jupyter Notebook в середовищі Google Colaboratory.

Веб-скрапінг (витягання веб-даних) - це перетворення інформації з веб-сторінок у структуровані дані. Як правило, виконується за допомогою комп'ютерних програм, що завантажують веб-сторінку напряму через HTTP-протокол, або запускають і керують веб-браузером, на подобі людини. Ця частина зосереджена на бібліотеках, які завантажують веб-сторінки з сервера через HTTP-протокол. З допомогою них, маючи лише URL-адреси, можна швидко зібрати великий масив даних зі сторінок з однакою структурую, для подальшого опрацювання. Ми будемо використовувати **Pandas DataFrame** для зберігання зібраних даних.

Поширені сфери використання:

- порівняння цін: деякі сервіси використовують веб-скрапінг, щоб добувати дані з інтернет-магазинів та використовувати їх для порівняння цін на товари;

- збір адрес електронної пошти: багато компаній, які використовують електронну пошту, як засіб маркетингу, використовують веб-скрапінг для їх збору;

- збирання даних з соціальних медіа: часто веб-скрапінг використовують для отримання даних з соціальних медіа, як от Твітер, для відстеження світових теденцій у політиці, криптовалютах, спорті, тощо.

Проте, соціальні мережі, як от Фейсбук, вміють відслітковувати, коли операції здійнює 'робот', а не реальний користувач і блокувати їх, використовуючи капчу, або інші методи. Це може призвести до блокування акаунту;

- дослідження та розробка в різних сферах: збір даних з сайтів для різного роду аналізу, проведення досліджень, побудови прогностичних моделей, збору і побудови статистичної візуалізації, тощо (наприклад збір прогнозу погоди з різних джерел, документів, наукових статей, фінансових звітів, результатів тендерів);

- створення різного роду агрегаторів інформації: новин, вакансій, подій, тощо;

Інструменти для веб-скрапінгу

Так, як кожен веб-сайт має свою унікальну структуру документу, необхідно створювати унікальний веб-скрапер для кожного випадку. Проте, це окуповується, якщо потрібно отримати дуже багато інформації зі сторінок з однаковою структурою. Варто зауважити, що є платні сервіси-конструктори, які дозволяють створювати веб-скрапери без написання коду, проте їх функціональність обмежена.

Спершу потрібно 'вручну' проаналізувати веб-сторінку, для виявлення шаблонів розташування інформації, використавши консоль розробника. У ній провести інспекцію DOM-дерева сторінки і виявлення HTML-тегів, у яких міститься шукана інформація.

Консоль розробника відкривається одним із способів:

- натиснути праву кнопку миші на веб-сайті для відкриття контекстного меню, після чого обрати варіант 'Перевірити' або 'Inspect';

- натиснути комбінацію клавіш: Ctrl+Shift+i/F12/Command+Option+i в залежності від операційної системи та браузера;

Після цього - використати отриману інформацію про структуру сторінки і одну із описаних нижче бібліотек для добування даних і подальшого збереження.

Всі приклади використання бібліотек знаходяться за посиланням: shorturl.at/pqLM3

Newspaper3k

Newspaper3k - бібліотека створена спеціально для веб-скрапінгу новинних сайтів. Спосіб роботи з нею такий:

- завантажити сторінку, з якої потрібно зібрати дані, передавши у метод бібліотеки URL адресу

- використовувати функціональність бібліотеки, для аналізу і добування даних із даної сторінки

Посилання на офіційну документацію : https://newspaper.readthedocs.io/en/latest/

BeautifulSoup + Requests library

Newspaper3k працює добре для новинних сайтів, проте помилки все ж присутні, якщо проаналізувати колонку `authors` з запасника з кодом, видно, що часто вона не є заповненою, або заповнена неправильно. Коли ж потрібно розпарсити не новинний сайт, або мати дуже велику гнучкість при роботі з новинним, знадобляться наступні бібілотеки.

Requests - HTTP бібліотека для мови Python. Вона реалізовує різноманітні HTTP запити і їх обробку.

BeautifulSoup - це бібліотека Python для вилучення даних з файлів HTML та XML. Фактично ми будемо використовувати бібліотеку **Requests** для того, щоб завантажити HTML сторінку, після чого розпарсити її бібліотекою **BeautifulSoup**.

Документація:
https://beautiful-soup-4.readthedocs.io/en/latest/
https://docs.python-requests.org/en/latest/

Requests не працюватиме з SPA (Single Page Application). Оскільки їх принцип - підвантажити 'пусту' HTML сторінку з JavaScript кодом з сервера, після чого він (JavaScript код) побудує решту сторінки. Відповідно, оскільки ми не отримуємо повну HTML сторінку з сервера, **BeutifulSoup** не матиме що парсити. Щоб вирішити цю проблему можна використати бібліотеку **Selenium**, про яку ітиметься мова в наступній частині.

Також в наступній частині буде розгялнуто роботу з **Scrapy** та **lxml.**

Завдання:

- використовуючи приклади коду, які знаходяться за посиланням вище, написати код для скрапінгу певної веб-сторінки. Використовувати

бібліотеки **Newspaper3k** або **BeautifulSoup + Requests.** У випадку, якщо скрапер писатимете у об'єктно-орієнтованому стилі, використати клас ScraperFactory;

- використавши написайний у попередньмоу завданні скрапер, зібрати дані з кількох таких сторінок і розмістити їх у **pandas DataFrame** (мінімум 5);

- навести приклади SPA сторінок, які неможливо розпарсити, використовуючи **BeautifulSoup + Requests** (мінімум 1);